

Actividad UT3- 1

- Seguir las pautas de nomenclatura indicadas en clase
- Asociar en todos los casos un icono a la aplicación y a los formularios
- Evitar entregar proyectos con advertencias de SonarLint
- Tiempo estimado: **6 horas**

En esta actividad vamos a desarrollar un **gestor de imágenes**, con una interfaz similar a la de la siguiente imagen:



La funcionalidad de los distintos botones es la siguiente:

- *Seleccionar directorio*: permite seleccionar un directorio del que se cargarán las imágenes en el visor. Solamente se cargarán archivos JPG.
- *Analizar actual*: analiza la imagen actual para deducir su título, categoría y si es o no una imagen inapropiada.
- *Analizar todas*: analiza todas las imágenes cargadas.
- *Clasificar actual*: clasifica la imagen en disco en función de su categoría y subcategoría. La imagen se copiará al directorio correspondiente a su categoría, dentro del directorio del que se han cargado las imágenes.
- *Clasificar todas*: clasifica todas las imágenes cargadas.

Tened en cuenta las siguientes indicaciones:

- El único botón que deberá estar habilitado al iniciar la aplicación es *Seleccionar directorio*. El resto se habilitarán cuando haya alguna imagen cargada en el visor. (**Trigger**)
- Al pasar el ratón por encima de un botón habilitado, el texto del mismo se pondrá en negrita. (**Trigger**)
- Cuando una imagen sea catalogada como inapropiada, deberá aparecer desenfocada. (**Trigger**)
- Cuando una imagen sea catalogada como inapropiada, aparecerá una imagen de aviso en la parte superior izquierda. (**Trigger**)
- El menú deberá aparecer cuando el ratón entre en él, y ocultarse cuando el ratón salga. (**Trigger**)
- La información de la imagen (ruta del fichero, título, categoría y si es inapropiada) se obtendrá a partir de un objeto de la clase *Imagen*. (**Binding**)
- El título asociado a la imagen se almacenará en el objeto tal y como lo suministra la API (en minúsculas), pero en la interfaz aparecerá con la primera letra en mayúsculas. (**Convertidor**)
- La categoría asociada a la imagen se almacenará en el objeto tal y como la suministra la API (utilizando el separador _ entre categoría y subcategoría),

pero en la interfaz aparecerá con el separador / en caso de ser necesario.
(**Convertidor**)

- Durante el análisis de una imagen se cambiará el icono del cursor para mostrar al usuario la duración del proceso.
- Durante el análisis de todas las imágenes, se indicará el progreso en el propio botón.
- Es necesario gestionar las excepciones que se puedan producir, tanto al manipular el sistema de ficheros como al utilizar la API.
- Informar al usuario utilizando *MessageBox* de las posibles excepciones. También informar cuando se complete una tarea que implique una llamada a la API.

¿Qué tengo que aprender?

⇒ Clases para el manejo del sistema de ficheros (*Directory*, *File* y *Path*).

<https://docs.microsoft.com/en-us/dotnet/api/system.io.directory?redirectedfrom=MSDN&view=netframework-4.8>

<https://docs.microsoft.com/en-us/dotnet/api/system.io.file?view=netframework-4.8>

<https://docs.microsoft.com/en-us/dotnet/api/system.io.path?view=netframework-4.8>

⇒ Cómo se utiliza el *FolderBrowserDialog*

Ejemplo en Aules

⇒ Cómo se utiliza la API *ComputerVision*

Ejemplo en Aules

EVALUACIÓN

Diseño del layout <i>La interfaz de la aplicación es similar a la mostrada en el enunciado. El comportamiento al redimensionar la ventana se ajusta al ejemplo suministrado. Se ha configurado correctamente el título de la ventana y el icono de la ventana y de la aplicación.</i>	10 puntos
Carga de imágenes <i>Se utiliza el control FolderBrowserDialog para seleccionar el directorio. Se controla si el usuario cancela la selección. Se filtra el tipo de fichero a cargar a JPG. Se muestran todas las imágenes en el visor y se inicializan correctamente el número de imagen actual y el número total.</i>	15 puntos
Análisis de imágenes <i>Se utiliza la API Computer Vision para obtener información de las imágenes. Si se analiza la actual se actualiza el cursor, y si se analizan todas se muestra el progreso en el botón. Se muestra la información actualizada en la imagen utilizando las conversiones adecuadas. Se informa al usuario del resultado de la acción.</i>	15 puntos
Clasificación de imágenes <i>Si es necesario, se crean los directorios y subdirectorios correspondientes a la categoría y subcategoría de la imagen. Se copian los ficheros adecuados en las carpetas de las categorías. Se informa al usuario del resultado de la acción. No aparecen rutas en el código (hardcode).</i>	15 puntos
Imágenes inapropiadas <i>Si la imagen es catalogada como inapropiada se le aplica un efecto de desenfoque. También se muestra el icono de aviso correspondiente.</i>	15 puntos
Menú y botones de desplazamiento <i>El menú de opciones tiene el efecto de animación indicado. Los botones están deshabilitados si no hay imagen en el visor. Al pasar el ratón por encima de los botones habilitados se resalta el texto del botón. Los botones de desplazamiento no permiten respetar el número de imágenes cargadas.</i>	15 puntos
Gestión de excepciones <i>Se controlan las excepciones al manipular el sistema de ficheros y al utilizar la API. Se informa al usuario en caso de que se produzca una excepción.</i>	10 puntos
Estructura del código, nomenclatura y SonarLint <i>Se utilizan los recursos del lenguaje de programación para estructurar el código correctamente. Se siguen las reglas de nomenclatura indicadas en clase para los distintos elementos del lenguaje. El código no contiene avisos de SonarLint.</i>	5 puntos
TOTAL	100 puntos